Robotic Exploration of the Icy Moons of the Gas Giants.

Patrick H. Stakem

Number 33 in Space Series
Edition - 1

Table of Contents

Introduction

As we stretch our exploration of our solar system, it gets much more challenging. This is due to the distances involved, and the vast diversity of the places we are visiting. W have issues of power and communication as we reach further out to our neighbors. We are seeking answers to questions such as, are we alone here? Can we live here? Are there materials we can use?

Due both to their numbers and their diversity, exploration of the moons of Jupiter and Saturn will be costly and difficult. Due to distance, the explorers will be pretty much on their own. The targets will have to be chosen carefully, with a priority on those potentially harboring life. The icy moons are of interest, since they have water under an ice crust. Water is essential to life as we know it.

This book presents an approach to the exploration of the moons of Jupiter and Saturn.

Author

The author received a Bachelors degree in Electrical Engineering from Carnegie-Mellon University, and Masters Degrees in Physics and Computer Science from the Johns Hopkins University.

He was glued to the black & white TV for the launch of the Vanguard, the U. S.'s first satellite, through the Apollo missions.

He began his career in Aerospace with Fairchild Industries on the ATS-6 (Applications Technology Satellite-6), program, a communication satellite that developed much of the technology for the TDRSS (Tracking and Data Relay Satellite System). At Fairchild, Mr. Stakem made the amazing discovery that computers were put onboard the spacecraft. He quickly made himself the expert on their support. He followed the ATS-6 Program through its operation phase, and worked on other projects at NASA’s

Goddard Space Flight Center including the Hubble Space Telescope, the International Ultraviolet Explorer (IUE), the Solar Maximum Mission (SMM), some of the Landsat missions, and others. He was posted to NASA's Jet Propulsion Laboratory for the MARS-Jupiter-Saturn (MJS-77), which later became the Voyager mission. It is still operating and returning data from outside the solar system at this writing.

The author has been at almost all of the NASA facilities and launch sites in an official capacity. There is always time to slip away and visit the museums.

Mr. Stakem was affiliated with the Whiting School of Engineering of the Johns Hopkins University. He received NASA's Space Shuttle Program Managers commendation award, as well as two NASA Group Achievement awards, the NASA Apollo-Soyuz Test Program Award, and a Certificate of Appreciation, NASA Earth Science Technology Office. Mr. Stakem has completed over 42 NASA Certification Courses in various areas,

He participated in tow NSA-sponsored Summer Engieering Boot Camps for college students, resulting in a large autonomous tracked robot being deployed to Greenland to measure ice thickness. He arranged two summer sessions at a local Engineeing college, involving Brazilian students under their Scientific Mobility Program. Theses programs focused on Earth and Interplanetary Cubesdat Missions.

The Gas Giants

The Gas giants are the planets Jupiter and Saturn. Exploration of these are the responsibility of the Jet Propulsion Laboratory. These each have extensive ring and moon systems that have been imaged, but not yet explored.

Pioneer 10 was the first mission to Jupiter, followed by Pioneer-11

in 1973, and, as of this writing, there have been 8 total. Jupiter has a very high trapped radiation environment. The moons are mostly all different, and some are thought to be capable of hosting life, as we know it. The moon Io has volcanic activity, and Europa has water ice on the surface. Europa is considered “one of the most promising extraterrestrial habitable environments in our solar system” according to the most recent Planetary Society's Decadal Survey.

NASA is looking at the Explorer CubeSat for Student Involvement in Travels to Europa (ExCSITE) Mission. This involves multiple Cubesat imagers and impactors.

The Voyager missions to Jupiter and Saturn were originally named the “Grand Tour” and were to have visited Mars, Jupiter, and Saturn, with possibly some of the outer planets as well. The mission was called MJS-77. Budget constraints caused the mission to refocus on Jupiter and Saturn alone. The author worked on this mission.

To the outer planets, the Gas and Ice Giants, we hitch a ride on the Interplanetary Superhighway, now superseded by the Interplanetary Transport Network. This is simply a series of optimized, gravitationally-determined paths through the solar system, that are energy (read: propellant) efficient, but can result in long durations. The path takes advantage of gravity assists from various planets, and what are called low energy transfers.

Jupiter

Jupiter has 79 known moons, as of this writing, and perhaps 1 million Trojans of 1 kilometer or larger. These tend to congregate at the L4 and L5 Lagrange points. The largest has a diameter of several hundred kilometers. The discussed mission will undoubtably discover additional moons. The one way light time for Jupiter is 33-53 minutes. Earth has to be on the same side of the

Sun as Jupiter for the communications to work.

Jupiter has all sorts of moons. The 4 largest were seen by Galileo around 1610. The moon Io is known to have 400 active volcanos, and may have a magnetic field. It has a thin atmosphere of sulphur dioxide, and is embedded in the radiation field of Jupiter. Moons are continuously discovered, and currently 7 are on the lost list.

Europa has a layer of water ice, and it is postulated there is an ocean under the ice. This might be a habitat for life forms. Similar to the exploration of the environment under Antarctic ice, any probe would have to drill its way down to liquid water.

Ganymede is made up of rock and water-ice, with a possible salt water ocean under the ice. There is a thin oxygen atmosphere.

Callisto has a rather thin carbon dioxide atmosphere. It may host a water ocean. This would also be a good place to look for life. The moon is out of Jupiter's intense tapped radiation fields, and has been considered a a base of operations when we get around to sending Astronauts to Jupiter.

I just mentioned details of four of the Jovian moons, the Galilean ones. The amount of diversity is a challenge to exploration. We will need ground explorers, flying explorers, and sub-ice vehicles.

The Cassini spacecraft observed the planet from close-up in the year 2000, and studied the atmosphere. Galileo entered Jupiter orbit in 1995, and returned data on the planet and the four Galilean moons until 2003. Three of the moons have thin atmospheres, and may have liquid water. The moon Ganymede has a magnetic field. Galileo was also in the right place at the right time to see the comet Shoemaker-Levy-9 enter the Jovian atmosphere, and launched an atmospheric probe.

The Juno mission to Jupiter arrived after 5 years of travel, and was getting settled in to begin its observations. This project was

launched in August of 2011, and arrived at Jupiter in July 2016. It was placed in Jupiter elliptical polar orbit for 5 years, and had its mission extended to 2021. Then it will enter the Jovian atmosphere and burn. This is to avoid any biological contamination of Jupiter or its moons. The orbit was chosen to minimize contact with Jupiter's intense trapped radiation belts. It's sensitive electronics are housed in "the Juno Radiation vault," with 1 cm titanium walls. It has available to it some 420 watts of power, from the solar arrays.

The spacecraft weighs over 1,500 kg. It uses 3 efficient solar panels of 2.7 x 8.9 meters long These will be exposed to about 4% of the sunlight at Earth. It left Florida on an Atlas-V vehicle. The perijove, or closest distance to the planet was planned to be 4,200 km. The highest altitude at apojove is 8.1 million kilometers.

The spacecraft includes infrared and microwave instruments to measure the thermal radiation from Jupiter's atmosphere, being particularly interested in convection currents. It's data will be used to measure the water in Jupiter's atmosphere, temperature and composition, and track cloud motions. The mission will also map Jupiter's magnetic and gravity fields. It is expected to probe the magnetosphere in the polar regions and observe the auroras.

Communications with Earth uses X-band to support 50 Mbps of data. The spacecraft will be constrained to 40 Mbytes of camera data per 11-day orbit period.

Juno uses a bi-propellant propulsion system for orbital insertion.

Saturn

Saturn and it's 82 known moons has a one-way light time around 1.4 hours. Saturn has been visited by spacecraft four times. The first was a flyby by Pioneer-10 in 1979. This showed the temperature of the planet was 250 degrees K. Voyager-1 visited in 1980. It conducted a close flyby of the moon Titan to study its atmosphere. It is, unfortunately, opaque in visible light. We do

know it rains methane. Voyager-2 swung by a year later, and data showed changes in the rings since its sister mission visited the year before. Temperature and pressure profiles of the atmosphere were gathered. Saturn's temperature was measured at 70 degrees above absolute zero at the top of the clouds, and -130 c near the surface. The flybys discovered additional moons, and small gaps in the rings. Saturn has relatively weak radiation belts.

Cassini was the fourth spacecraft to study Saturn, which has rings, although smaller than Jupiter. The rings were confirmed by the Voyager spacecraft in the 1980's. The one-way communications time varies form 68-84 minutes. It has also collected data on the Saturnian moons Titan, Enceladus, Mimas, Tethys, Dione, Rhea, Iapetus, and Helene. Things are strange in the Saturnian system. Cassini observed a hurricane in 2006 on the planet's south pole. It appears to be stationary, 5,000 miles (8,300 km) across, 40 miles (67 km) high, with winds of 350 mph (560 kph). The large moon Titan has lakes of a liquid hydrocarbon, with possible seas of methane and ethane. Cassini launched a probe *Huygens* to Titan, and it landed on solid ground below the atmosphere. The Cassini mission was responsible for the discovery of seven new moons. Cassini entered Saturn's atmosphere and was incinerated, in 2017

Cassini observed a massive storm on Saturn, the great white spot, that recurs every 30 years. The storm, larger than the red one on Jupiter, exhibited a discharge that spiked the temperature 150 degrees. At the same time, Earth observations showed a large increase in atmospheric ethylene gas. It also discovered large lakes or seas of hydrocarbons near the planet's north pole.

The Cassini mission discovered a possible atmosphere on the moon Enceladus, with ionized water vapor, and ice geysers. Many of the Saturnian moons are in tidal lock with their mother planet. Being so close to its giant neighbor Jupiter affects the Saturnian system. Titan, second only to Jupiter's Ganymede in size, has an Earth-like atmosphere, evidence of water flow, and dry lake beds.

Cassini cost $3 billion ($10^9$) and couldn't get close enough to Saturn's rings because of the ice particles. One hit could cancel the entire mission. Imagine instead a Cassini-size spacecraft at a safe distance, deploying swarms of hundreds of Cubesats. The moons Janus and Epimetheus are about the same size, are almost in the same orbit. They swap orbits every 4 years.

The Saturnian moon Mimas has a visible impact crater, as do may of its sister moons. Enceladus emits geysers of water vapor and dust. The moon Dione shows a past history of tectonic activity. The moon Titan has an atmosphere denser than Earth's, and body's of liquid water on its surface.

To define a term here, Saturn also has moonlets, which are embedded in the rings.

Titan, one of Saturn's Moons, is larger than the planet Mercury. It seems to have a nitrogen-rich, Earth-like atmosphere. The moons are divided into ten orbital groups, which means they could in theory be explored together. In the Ring system, shepherding moons influence the rings and the gaps between them.

The Challenge

For the Moon, Venus, or Mars, we can send a lander/rover, usually two for redundancy. The one-way light time is not too bad, and we can return volumes of data. We look forward to colonizing the Moon and Mars, but we still have un-answered questions. While we are working on that, we set our sights further out. The asteroid belt, beyond Mars, has some half-a-mission chunks of rock that we might find interesting. The shear number of targets to explore basically demands a different approach to exploration. Did I mention that some of the asteroids have one or more moons?

Working with brilliant grad- and undergrad students, the author explored different options. In a paradigm shift, they defined a mission of a carrier vehicle, or Mothership that could hold and

deploy hundreds of independent explorers. These simple explorer units, based on the Cubesat architecture, would return data to the mothership for forwarding to Earth, which would not necessarily be reachable all the time. So, the Mothership would be a store-and-forward node. At the same time, the mothership would have to take responsibility for locating targets of opportunity, deploying appropriate explorers, and implement command and control locally. Like the orbiting explorers of Jupiter, Saturn, Uranus, Neptune, and Pluto, they are on their own, due to communication availability and achievable bandwidth.

We evolved a concept called, “Fly the Control Center,” for a level of on-site autonomy. This would use standard open-source control center software, augmented by a Agent, that could analyze conditions, make decisions, and take action independently. It would also be able to learn. Engineering data would be stored in a database, and trended. It would be sent back to Earth as bandwidth and visibility allowed. The collected science data would be partially processed on the Mothership, to reduce download bandwidth requirments.

This book expands on the ANTS concept, a Juno-alternative and Asteroid Belt explorer concept to address the exploration of the Icy moons of Jupiter and Saturn.

Exploring the Gas Giants

The Gas giants are the planets Jupiter and Saturn. These are the responsibility of the Jet Propulsion Laboratory. These each have extensive ring and moon systems that have been imaged, but not yet explored.

Pioneer 10 was the first mission to Jupiter, followed by Pioneer-11 in 1973, and, as of this writing, there have been 8 total. Jupiter has a very high trapped radiation environment. They are mostly all different, and some are thought to be capable of hosting life, as we know it. The moon Io has volcanic activity, and Europa has water

ice on the surface. Europa is considered “one of the most promising extraterrestrial habitable environments in our solar system” according to the most recent Planetary Society's Decadal Survey. A proposed mission, ExCSITE, would provide characterization of the surface properties.

NASA is looking at the Explorer CubeSat for Student Involvement in Travels to Europa (ExCSITE) Mission. This involves multiple Cubesat imagers and impactors.

The Voyager missions were originally terms the “Grand Tour” and were to have visited Mars, Jupiter, and Saturn, with possibly some of the outer planets as well. The mission was called MJS-77. Budget constraints caused the mission to refocus on Jupiter and Saturn alone. The author worked on this mission.

There was a NASA mission proposed termed SARA – Saturn Autonomous Ring Array, which involved the ANTS Concept. Besides the moons of the Ice Giants, a similar arrangement, with different ops concepts could be used with Cubesats.

Jupiter

Jupiter has 79 known moons, and perhaps 1 million Trojans of 1 kilometer or larger. The International Astronomical Union has just announced the discovery of 12 previously unknown moons of Jupiter, by an observatory high in the Andes in Chile. Only one has been named so far, (Valetudo, a great-granddaughter of Jupiter). and 600,000 known Trojans.

The International Astronomical Union is in charge of name selection for planets, moons, asteroids, comets, and such. Generally, the discoverer submits their suggestion, which is usually confirmed.

These tend to congregate at L4 and L5. The largest has a diameter

of several hundred kilometers. The International Astronomical Union just announced as this book was being updated the discovery of 12 previously unknown moons of Jupiter, by an observatory high in the Andes in Chile. Only one has been named so far, Valetudo, a great-granddaughter of Jupiter The one way light time for Jupiter is 33-53 minutes.

The Cassini spacecraft observed the planet from close-up in the year 2000, and studied the atmosphere. Galileo entered Jupiter orbit in 1995, and returned data on the planet and the four Galilean moons until 2003. Three of the moons have thin atmospheres, and may have liquid water. The moon Ganymede has a magnetic field. Galileo was in the right place at the right time to see the comet Shoemaker-Levy-9 enter the Jovian atmosphere, and launched an atmospheric probe.

The Hubble Space Telescope discovered that the satellite of Jupiter, Ganymede, has a large saltwater ocean, under an ice crust. This is now thought to be the best current hope for finding life on another planet.

The Juno mission to Jupiter has just arrived after 5 years of travel, and was getting settled in to begin its observations. This project was launched in August of 2011, and arrived at Jupiter in July 2016. It was placed in Jupiter elliptical polar orbit for 5 years, and will de-orbit into Jupiter in February 2018. This is to ensure burn-up of the spacecraft to avoid any biological contamination of Jupiter or its moons. It is scheduled to make 37 orbits. The orbit was chosen to minimize contact with Jupiter's intense trapped radiation belts. It's sensitive electronics are housed in “the Juno Radiation vault,” with 1cm titanium walls. It will have available to it some 420 watts of power, from the solar arrays.

The spacecraft weighed over 1,500 kg. It uses 3 solar panels of 2.7 x 8.9 meters long These are exposed to about 4% of the sunlight at Earth. It left Florida on an Atlas-V vehicle. The perijove, or closest distance to the planet was planned to be 4,200 km. The

highest altitude at apojove is 8.1 million kilometers.

It includes infrared and microwave instruments to measure the thermal radiation from Jupiter's atmosphere, being particularly interested in convection currents. It's data will be used to measure the water in Jupiter's atmosphere, and measure atmospheric temperature and composition, and track cloud motions. The mission will also map Jupiter's magnetic and gravity fields. It is expected to probe the magnetosphere in the polar regions and observe the auroras.

Saturn

Saturn and it's 82 known moons has a one-way light time around 1.4 hours. Saturn has been visited by spacecraft four times. The first was a flyby by Pioneer-10 in 1979. This showed the temperature of the planet was 250 degrees K. Voyager-1 visited in 1980. It conducted a close flyby of the moon Titan to study its atmosphere. It is, unfortunately, opaque in visible light. We do know it rains methane. Voyager-2 swung by a year later, and data showed changes in the rings since its sister mission visited the year before. Temperature and pressure profiles of the atmosphere were gathered. Saturn's temperature was measured at 70 degrees above absolute zero at the top of the clouds, and -130 C near the surface. The flybys discovered additional moons, and small gaps in the rings.

Cassini was the fourth spacecraft to study Saturn, which has rings, and is smaller than Jupiter. The rings were confirmed by the Voyager spacecraft in the 1980's. Cassini entered into Saturnian orbit, and is still returning data. The one-way communications time varied from 68-84 minutes. It has also collected data on the Saturnian moons Titan, Enceladus, Mimas, Tethys, Dione, Rhea, Iapetus, and Helene. Enceladus has definite organic compounds, nitrogen and oxygen based, in plumes.

Things are strange in the Saturnian system. Cassini observed a hurricane in 2006 on the planet's south pole. It appears to be

stationary, 5,000 miles (8,300 km) across, 40 miles (67 km) high, with winds of 350 mph (560 kph). The large moon Titan has lakes of a liquid hydrocarbon, with possible seas of methane and ethane. Cassini launched a probe *Huygens* to Titan, and it landed on solid ground below the atmosphere. The Cassini mission was responsible for the discovery of seven new moons of Saturn.

Cassini observed a massive storm on Saturn, the great white spot, that recurs every 30 years. The storm, larger than the red one on Jupiter, exhibited a discharge that spiked the temperature 150 degrees. At the same time, Earth observations showed a large increase in atmospheric ethylene gas. It also discovered large lakes or seas of hydrocarbons near the planet's north pole.

Cassini discovered a possible atmosphere on the moon Enceladus, with ionized water vapor, and ice geysers. Many of the Saturnian moons are in tidal lock with their mother planet. Being so close to its giant neighbor Jupiter affects the Saturnian system.

Cassini cost \$3 billion ($\$10^9$) and couldn't get close enough to Saturn's rings because of the ice particles. One hit could have ended the mission.

Twenty of the moons were added in 2019, thrusting Saturn ahead of Jupiter for having the most moons. Not all have been named. We're starting to run out of Roman and Greek gods, so the new moons will be Norse, Gallic, and Inuit groups. Seventeen of them are in retrograde orbits; they move opposite of Saturn's rotation. The discovery is credited to the Subaru telescope in Hawaii.

The moon Enceladus has definite organic compounds, nitrogen and oxygen based, in plumes.

Technology readiness levels, by NASA

Technology Readiness Levels are defined to show the readiness of a certain technology to be used, NASA's definition extends to

“ready to be used at the specified place”. A newly released technology has a TRL of 0. The numbers extend to 7, meaning having been flown successfully in space.

1. Basic principles observed and reported
This is the lowest "level" of technology maturation. At this level, scientific research begins to be translated into applied research and development.

2. Technology concept and/or application formulated
Once basic physical principles are observed, then at the next level of maturation, practical applications of those characteristics can be 'invented' or identified. At this level, the application is still speculative: there is not experimental proof or detailed analysis to support the conjecture.

3. Analytical and experimental critical function and/or characteristic proof of concept.

At this step in the maturation process, active research and development (R&D) is initiated. This must include both analytical studies to set the technology into an appropriate context and laboratory-based studies to physically validate that the analytical predictions are correct. These studies and experiments should constitute "proof-of-concept" validation of the applications/concepts formulated at TRL 2.

4. Component and/or breadboard validation in laboratory environment.

Following successful "proof-of-concept" work, basic technological elements must be integrated to establish that the "pieces" will work together to achieve concept-enabling levels of performance for a component and/or breadboard. This validation must be devised to support the concept that was formulated earlier, and should also be consistent with the requirements of potential system applications.

The validation is "low-fidelity" compared to the eventual system: it could be composed of ad hoc discrete components in a laboratory

TRL's can be applied to hardware or software, components, boxes, subsystems, or systems. Ultimately, we want the TRL level for the entire systems to be consistent with our flight requirements. Some components may have higher levels than needed.

5. Component and/or breadboard validation in relevant environment.

At this level, the fidelity of the component and/or breadboard being tested has to increase significantly. The basic technological elements must be integrated with reasonably realistic supporting elements so that the total applications (component-level, sub-system level, or system-level) can be tested in a 'simulated' or somewhat realistic environment.

6. System/subsystem model or prototype demonstration in a relevant environment (ground or space).

A major step in the level of fidelity of the technology demonstration follows the completion of TRL 5. At TRL 6, a representative model or prototype system or system - which would go well beyond ad hoc, 'patch-cord' or discrete component level bread-boarding - would be tested in a relevant environment. At this level, if the only 'relevant environment' is the environment of space, then the model/prototype must be demonstrated in space.

7. System prototype demonstration in a space environment.

TRL 7 is a significant step beyond TRL 6, requiring an actual system prototype demonstration in a space environment. The prototype should be near or at the scale of the planned operational system and the demonstration must take place in space.

The TRL assessment allows us to consider the readiness and risk of

our technology elements, and of the system.

A new Approach

I will now discuss a new potential approach to addressing the complexity of exploring the moons of the Gas Giants. A whole new paradigm is required to address the complexity of this mission, but at the same time, the payoff will be massive if life is discovered.

Enabling Technologies

This section will discuss some of the enabling technologies that will be used in the Cubesat Swarm mission definition. Use of existing technologies from previous missions was a goal. An ops concept will be presented later.

Use of a common hardware bus and software architecture for all swarm members, to the greatest extent possible, is a goal. Only the sensor sets will be unique. A Cubesat model for the hardware, and NASA GSFC's Core Flight Software is baselined. A standard linux software operating environment and database will be used.

Individual units will be organized as a swarm, a collection of functional units that is self-organizing,

Each member of the swarm will be aware visually of other swarm members in close proximity. This will be facilitated by having the Mothership as the center of the coordinate system. It will determine its position by celestial navigation. The Cubesats will have a similar capability. The mothership will maintain, as part of its onboard database, the location of all other members. It will also monitor for pending collisions and warn the participants. There will be rules concerning how close swarm members can get to each other, a virtual zone of exclusion. All Earth-based interaction with the swarm will be through the Mothership. Due to varying communication delays, operation of the swarm by teleoperation from Earth is not feasible. The Swarm could be on the opposite

side of the Sun from the Earth for extended periods. This is addressed by building autonomy into the system, and a large amount of non-volatile storage on the Mothership will be included for storage and archiving of science data. Earth is several hours away, in communication time.

In this concept, the Cubesats are the primary payload. The Mothership can be thought of as a very large Cubesat. The architecture is kept as close as possible.

Use of a common hardware bus and software architecture for all swarm members, to the greatest extent possible, is a goal. Only the sensor sets will be unique. A Cubesat model for the hardware, and NASA GSFC's Core Flight Software is baselined. A standard linux software operating environment and database will be used.

Each swarm member will be equipped with one or more cameras, not only for target investigation, but also for observing the position and relative motions of other swarm members.

Using standard linux clustering software (Beowulf), the Mothership and undeployed swarm members will be able to form an ad-hoc cluster computer to process science data in-situ.

LAN-based Mesh network software will be used. The Mothership's main computer will be a Raspberry-Pi based cluster, of probably 64-nodes. Radiation issues are addressed by a vault, similar to Juno's, and Rad-Hard Software, discussed later, running on a Rad-Hard microcontroller.

Complexity in a system generally derives from two parameters, the number of units, and the number of interactions. A swarm of Cubesats is complex, compared to a single spacecraft. This is somewhat balanced by the relative simplicity of the units, their standardization, flexibility, and redundancy.

Cube-Xs

Cube-X is a concept derived from Cubesats. A Cube-X can be as simple as a Cubesat sitting on a mobility platform. “Cubesat” is taken here as a general concept for the electronics of a mobile sensor platform.The mobility platform can be wheels, tracks, legs, quad-copter-like, water vessel, or submarine. The individual explorers are considered expendable, but will be recovered if feasible.

Cubesats

A Cubesat is a small, affordable satellite that can be developed and launched by college, high schools, and even individuals. The specifications were developed by Academia in 1999. The basic structure is a 10 centimeter cube, (volume of 1 liter) weighing less than 1.33 kilograms. This allows multiples of these standardized packages to be launched as secondary payloads on other missions. A Cubesat dispenser has been developed, the Poly-PicoSat Orbital Deployer, P-POD, that holds multiple Cubesats and dispenses them on orbit. They can also be launched from the Space Station, via a custom airlock. ESA, the United States, Russia, and several other countrys provide launch services. The Cubesat origin lies with Prof. Twiggs of Stanford University and was proposed as a vehicle to support hands-on university-level space education and opportunities for low-cost space access. This was at a presentation at the University Space Systems Symposium in Hawaii in November of 1999.

Build costs can be lower than $10,000, with launch costs ranging around $100,000, a most cost-effective price for achieving orbit. This approach is termed ride-sharing. +The low orbits of the Cubesats insure eventual reentry into the atmosphere, so they do not contribute to the orbital debris problem.

Central to the Cubesat concept is the standardization of the interface between the launch vehicle and the spacecraft, which allows developers to pool together for launch and so reduce costs

and increase opportunities. As a university-led initiative, Cubesat developers have advocated many cost-saving mechanisms, namely:

- A reduction in project management and quality assurance roles .
- Use of student labor with expert oversight to design, build and test key subsystems.
- Reliance on non-space-rated Commercial-Off-The-Shelf (COTS) components .
- Limited or no built-in redundancy (often compensated for by the parallel development of Cubesats) .
- Access to launch opportunities through standardized launch interfaces.
- Use of amateur communication frequency bands and support from amateur ground stations.
- Simplicity in design, architecture and objective .

Multiple Cubesats can be carried as secondary payloads on military and commercial flights. Cubesats, as small, inexpensive units have a higher mission risk tolerance.

Since the initial proposal of the concept, further efforts have been made to define internal and external interfaces made by various developers of Cubesat subsystems, products, and services that have defined the Cubesat 'standard' as it is today. A core strength of the Cubesat is its recognition of the need for flexibility in the definition of standards, and since conception, the standard has evolved to ensure that these design rules are as open as possible. The most significant of these further advances in definition have been for the POD systems (in order to meet launch requirements) and the modularization of the internal electronics.

A simple Cubesat flight controller can be developed from a standard embedded computing platform such as the Arduino. The

lack of radiation hardness can be balanced by the short on-orbit lifetime. The main drivers for a Cubesat flight computer are small size, small power consumption, wide functionality, and flexibility. In addition, a wide temperature range is desirable. The architecture should support a real time operating system, but, in the simplest case, a simple loop program with interrupt support can work.

Another important and related aspect in the design approach is that of modularity in a complete and integrated Cubesat life cycle, effectively representing a modular system of systems. The accelerated life cycle demonstrated consistently by small satellites, and harnessed by many Cubesat developers, can be further enhanced by the application of modularity to the complete life cycle.

A distributed target requires a distributed approach to exploration.

Overall Architecture

In the Icy Moons scenario, the Cubesats are the primary payload. The Mothership can be thought of as a very large Cubesat. The architecture is kept as identical as possible.

Each member of the swarm will be aware visually of other swarm members in close proximity. This will be facilitated by having the Mothership as the center of the coordinate system. The Cubesats will use the mothership as their reference. The mothership will maintain, as part of its onboard database, the location of swarm members.

Each swarm member will be equipped with one or more cameras, not only for target investigation, but also for observing the position and relative motions of other swarm members.

Using standard linux clustering software (Beowulf), the Mothership and undeployed swarm members will be able to form

an ad-hoc cluster computer to process science data in-situ. Within the Mothership, a LAN-based Mesh network software will be used.

Batteries and solar panels

The advanced batteries and solar panels on the ongoing Juno mission at Jupiter are functioning well. These would then be a candidate for the Swarm Mothership. Due to technology advances, solar cells can now be used out to 5 AU. The Mothership will have large arrays, but the individual Cubesats have limited area. They will be deployed fully charged. One area of interest is a deployable fabric solar panel, that can also be used as a solar sail.

Onboard Software

A robot is a special case of an embedded system. As such, it is generally more difficult to design, implement, and test. It must be designed carefully, because most of the Cubesat explorer functionality will rely on software, and the mission success will be directly related to software.

Software can be proprietary or Open Source, but almost all Cubesat software is open source.

Flight software has several distinguishing characteristics:

- There are no direct user interfaces such as monitor and keyboard. All interactions are through uplink and downlink.

- It interfaces with numerous flight hardware devices such as thrusters, reaction wheels, star trackers, motors, science instruments, temperature sensors, etc.

- It executes on radiation-hardened processors and microcontrollers that are relatively slow and memory-limited.

- It performs real-time processing. It must satisfy numerous

timing constraints (timed commands, periodic deadlines, async event response). Being late = being wrong.

- Besides attitude determination and control, the onboard embedded systems has a variety of housekeeping tasks to attend to.

NASA's Core Flight Executive, and Core Flight Software

The Core Flight Executive, from the Flight Software Branch at NASA/GSFC, is an open source operating system framework. The executive is a set of mission independent reusable software services and an operating environment. Within this architecture, various mission-specific applications can be hosted. The cFE focuses on the commonality of flight software. The Core Flight System (CFS) supplies libraries and applications. Much flight software legacy went into the concept of the cFE. It has gotten traction within the Goddard community, and is in use on many flight projects, simulators, and test beds (FlatSats) at multiple NASA centers, as well as functioning in on-orbit Cubesats, and LRO.

The cFE presents a layered architecture, starting with the bootstrap process, and including a real time operating system. At this level, a board support package is needed for the particular hardware in use. Many of these have been developed. At the OS abstraction level, a Platform support package is included. The cFE core comes next, with cFE libraries and specific mission libraries. Ap's habituate the 5th, or upper layer. The cFE strives to provide a platform and project independent run time environment.

The boot process involves software to get things going after power-on, and is contained in non-volatile memory. cFE has numerous boot loaders for various compute architectures, use in space applications. The real time operating systems can be any of a number of different open source or proprietary products, VxWorks

and RTEMS for example. This layer provides interrupt handling, a scheduler, a file system, and interprocess communication.

The Platform Support Package (PSP) is an abstraction layer that allows the cFE to run a particular RTOS on a particular hardware platform. There is a PSP for desktop pc's for the cFE. The cFE Core includes a set of re-usable, mission independent services. It presents a standardized Application Program Interface (API) to the programmer. A software bus architecture is provided for messaging between applications.

The Event services at the core level provides an interface to send asynchronous messages, telemetry. The cFE also provides time services.

Aps include a Health and Safety Ap with a watchdog. A housekeeping AP for messages with the ground, data storage and file manager aps, a memory checker, a stored command processor, a scheduler, a check-summer, and a memory manager. Aps can be developed and added to the library with ease.

The cFE has been released into the World-Wide Open Source community, and has found many applications outside of NASA.

NASA's software Architecture Review Board looked at the cFE in 2011. They found it a well thought-out product that definitely met a NASA need. It was also seen to have the potential of becoming a dominant flight software architectural framework. The technology was seen to be mature. NASA made the software available worldwide under a Open Source kicense, Developers are encouraged to submits compatible software back, to be evaluated and possibly added to the software library. This is how Open Source works, and how you can get your software into space.

The cFS is the core flight software, a series of aps for generally useful tasks onboard the spacecraft. The cFS is a platform and project independent reusable software framework and set of

reusable applications. This framework is used as the basis for the flight software for satellite data systems and instruments, but can be used on other embedded systems in general. More information on the cFS can be found at http://cfs.gsfc.nasa.gov/OSAL

The OS Abstraction Layer (OSAL) project is a small software library that isolates the embedded software from the real time operating system. The OSAL provides an Application Program Interface (API) to an abstract real time operating system. This provides a way to develop one set of embedded application code that is independent of the operating system being used. It is a form of middleware.

cFS aps

CFS aps are core Flight System (CFS) applications that are plug-in's to the Core Flight Executive (cFE) component. Some of these are discussed below.

CCSDS File Delivery (CF)

The CF application is used for transmitting and receiving files. To transfer files using CFDP, the CF application must communicate with a CFDP compliant peer. CF sends and receives file information and file-data in Protocol Data Units (PDUs) that are compliant with the CFDP standard protocol defined in the CCSDS 727.0-B-4 Blue Book. The PDUs are transferred to and from the CF application via CCSDS packets on the cFE's software bus middleware.

Limit check (LC)
The LC application monitors telemetry data points in a cFS system and compares the values against predefined threshold limits. When a threshold condition is encountered, an event message is issued and a Relative Time Sequence (RTS) command script may be initiated to respond/react to the threshold violation.

Checksum (CS)
The CS application is used for for ensuring the integrity of onboard memory. CS calculates Cyclic Redundancy Checks (CRCs) on the different memory regions and compares the CRC values with a baseline value calculated at system start up. CS has the ability to ensure the integrity of cFE applications, cFE tables, the cFE core, the onboard operating system (OS), onboard EEPROM, as well as, any memory regions ("Memory") specified by the users.

Stored Command (SC)
The SC application allows a system to be autonomously commanded 24 hours a day using sequences of commands that are loaded to SC. Each command has a time tag associated with it, permitting the command to be released for distribution at predetermined times. SC supports both Absolute Time tagged command Sequences (ATSs) as well as multiple Relative Time tagged command Sequences (RTSs).

Scheduler (SCH)
The SCH application provides a method of generating software bus messages at predetermined timing intervals. This allows the system to operate in a Time Division Multiplexed (TDM) fashion with deterministic behavior. The TDM major frame is defined by the Major Time Synchronization Signal used by the cFE TIME Services (typically 1 Hz). The Minor Frame timing (number of slots executed within each Major Frame) is also configurable.

File Manager (FM)
The FM application provides onboard file system management services by processing ground commands for copying, moving, and renaming files, decompressing files, creating directories, deleting files and directories, providing file and directory informational telemetry messages, and providing open file and directory listings. The FM requires use of the cFS application library.

OpSys

An *operating system* (OS) is a software program that manages computer hardware and software resources, and provides common services for execution of various application programs. Without an operating system, a user cannot run an application program on their computer, unless the application program is itself self-booting, and has an initiation module, that does the necessary hardware set-up.

For hardware functions such as input, output, and memory allocation, the operating system acts as an intermediary between application programs and the computer hardware, although the application code is usually executed directly by the hardware and will frequently call the OS or be interrupted by it. Operating systems are found on almost any device that contains a computer. The operating system functions need to be addressed by software (or possibly hardware), even if there is no entity that we can point to, called the Operating System. In simple, usually single-task programs, there might not be an operating system per se, but the functionality is still part of the overall software.

An operating system manages computer resources, including:

- Memory.
- I/O.
- Interrupts.
- Tasks/processes/application programs.
- File system
- clock.

The operating system arbitrates and enforces priorities. If there are not multiple software entities to arbitrate among, the job is simpler. An operating system can be off-the-shelf commercial or open source code, or the application software developer can decide to build his or her own. To avoid unnecessary reinvention of the

wheel an available product is usually chosen. Operating systems are usually large and complex pieces of software. This is because they have to be generic in function, as the originator does not know what application space it will be used in. Operating systems for desktop/network/server application are usually not applicable for embedded applications. Mostly they are too large, having many components that will not be needed (such as the human interface), and they do not address the real-time requirements of the embedded domain.

Adapting a commercial or open source operating system to a particular embedded domain can be tricky. Usually, the commercial operating systems need to be used "as-is" and the source code is not available. The software can usually be configured between well-defined limits, but there will be no visibility of the internal workings. For the open source situation, there will be a multitude of source code modules and libraries that can be configured and customized, but the process is complex. The user can also write new modules in this case.

Operating Systems designed for the desktop are not well suited for embedded These were developed under the assumption that whatever memory is required will be available, and real-time operation with hard deadlines is not required.

Real-time operating systems, as opposed to those addressing desktop, tablet, and server applications, emphasize predictability and consistency rather than throughput and low latencies. Determinism is probably the most important feature in a real-time operating system.

A microkernel operating system is ideally suited to embedded systems. It is slimmed down to include only those features needed, with no additional code. Barebones is the term sometimes used. The microkernel handles memory management, threads, and communication between processes. It has device drivers for only those devices present. The operating systems may have to be

recompiled when new devices are added. A file system, if required, is run in user space. MINIX, as an example of a streamlined kernel, has about 6,000 lines of code.

For computer systems like the Raspberry Pi, there are enough resources to run an operating system such as Linux. But linux is NOT a real time operating system. It can be patched to operate better in real-time applications, or a custom open source product such as FreeRTOS can be used.

For Arduino-based boards, the architecture is just getting sophisticated enough to run an operating system. In many cases, an operating system is not needed. Keep in mind, that some operating system functions still need to be implemented. Setting up the interrupt vectors at initialization is an example. The tasks that the computer has to do might be simple enough that a simple software loop architecture can do the job, perhaps with interrupts.

File Systems

Use of an industry-standard file system will ease the interface to storage and processing. There are several popular file systems, usually defined as part of a specific operating system. A file system provides a way to organized data, and file systems management services are part of the operating system. The operating systems may support several file formats. A file system organizes data. It presents a data-centric view of a digital storage system.

A file is a container of information, usually stored as a one-dimensional array of bytes. Historically, the file format and the nature of the file system were driven by the mechanism of data storage. On early computer tape units for mainframes, the access mechanism was serial, leading to long access times. With disk and solid-state storage, the access time was vastly improved, as the device is random access – the same access time applies for any data item.

Metadata includes information about the data in a file. This consists of the file name and type, and other parameters such as the size, date and time of creation, the data and time of last access, the owner and read/write/access permissions, when a backup was last made, and other related information.

A directory, like the manila file folder, is a special file that points to ("contains") other files. This allows files to be organized, and implements a hierarchical file system.

There are many file system standards. The Microsoft operating systems support the FAT and NTFS file systems among others. Linus supports those, as well as its own ext4. The FAT (File Allocation System) format originated with early support of 8-bit microprocessor systems. Fat-12 and FAT-16 had restrictions on the number of files in the root file system, but this has largely been removed with the introduction of FAT-32. File names are restricted to 8 characters, with a 3-character type specifier, the 8.3 format for file names. There are plenty of choices, there is no need to re-invent the wheel.

Onboard databases

Each member of the Swarm is self-documenting. It carries a copy of its Electronic Data Sheet (EDS) description, which can be updated. This defines the system architecture and capabilities, and has both fixed (as-built) and variable entries. The main computer in the Mothership has a copy of all of these, and can get updates by query. The Mothership also has parameters on each unit's state, such as electrical power remaining, temperature, position, etc. One value of the database is, if the Mothership needs a unit with a high resolution imager, it knows what unit that is, and whether it has been deployed or not. If it has been deployed, it will query the unit on its position and health status. Implementing the EDS in a true database has advantages, since the position of the data item in the database also carries information. It allows the use of off-the-shelf database tools. The individual Cubesats have a "light-weight" version of the database, while the Mothership has a more

sophisticated one. All the schema's are the same. The advantage of a formal database is the structure it imposes on the data

There are two parts of the tables, representing static and dynamic data. Static Data represents the hardware and software configuration of the swarm unit. These values are not expected to change during the unit's operation. The Dynamic Data table represents the sensors each particular unit has. These values can change, and the last values will be kept. Cubesats will exchange two types of data through their communication channel: primary observational data, along with secondary metadata which includes position and localization information along with timing as a part of the EDS during the mission.

The Mothership is responsible for aggregating all of the Cubesats' housekeeping and science data, and transmitting it back to Earth. This is also facilitated by the structure imposed by the database. An Open Source version of an SQL, such as MySQL, will be used The EDS documents will be in XML format

Data compression can be implemented onboard the Mothership , as well as preliminary data analysis for replanning.

Rad-Hard Software

This is a concept that implements routines that check and self-check, report, and attempt to re-mediate. It is an outgrowth of the testing and self-testing of a computers' functionality, with focus on detection of radiation induced damage. We know, for example, that one of the tell-tales for radiation damage is increasing current draw. At the same time, we monitor other activities and parameters in the system. This partially addresses the problem of operating with non-radiation hardened hardware in a high radiation environment. The baseline RaspberryPi has been radiation tested to 150 kRad, and was operational at that point. A rad-hard version is in development.

From formal testing results, and key engineering tools, we define likely failure modes, and possible remediation's. Besides self-test, we will have cross-checking of systems. Not everything can be tested by the software, without some additional hardware. First, we use engineering analysis that will help us define the possible hardware and software failure cases, and then define possible actions and remediation. None is this is new, but the approach is to collect together best practices in the software testing area, develop a library of RHS routines, and get operational experience. Another advantage of the software approach is that we can change it after launch, as more is learned, and conditions change.

Rad-Hard software runs in the background on the flight computer, or on a dedicated rad-hard device such as an Arduino. The software checks for the signs of pending failure from any known cause. The flight cpu monitors and trends current draw across the swarm, and takes critical action such as a reboot if it deems necessary. The Rad-Hard software will keep tabs on memory by conducting continuous CRC (cyclic redundancy checks). One approach to mitigating damage to semiconductor memory is “scrubbing,” where we read and write back each memory locations (being careful not to interfere with ongoing operations). This can be done by a background task that is the lowest priority in the system. Watchdog timers are also useful in getting out of a situation such as a Priority Inversion, or just a radiation-induced bit flip. There will be a pre-defined safe mode for the computer as well. Key state data from just before the fault will be stored. Unused portions of memory can be filled with bit patterns that are monitored for changes. We must be certain that all of the unused interrupt vectors point to a safe area in the code, so this will be reloaded periodically.

Functions within the RHS include current monitoring as a tell-tale of radiation damage, self-diagnosis suite, spurious interrupt test, memory test(s), checksums over code, data corruption testing, memory scrub, I/O functionality test, peripherals test, stack

overflow monitoring, and a watchdog timer. A complete failure modes and effects analysis will be conducted over the flight computer and associated sensors and mechanisms, and this will be used to scope the RHS. The systems will keep and report trending data on the flight electronics. In most cases, the only remediation is a reboot. However, since the units will have identical configurations, the data will be useful to be able to predict pending failures, and to possibly avoid and correct them. This will be used on the Mothership's and on the Cubesat's RaspberryPi-based flight computers. This provides a distributed fault detection and mitigation system, with learning.

We can also choose to implement a small, rad-hard recovery computer, which uses FRAM, which is fairly immune to radiation. The recovery computer would receive heart-beat signals from the cluster members onboard the mothership, and take recovery efforts if they are interrupted. A similar scheme could be used onboard the Cubesats, with little impact on size, weight, power, and cost. This would primarily be used to mitigate latchup.

Fly the Control Center

The Mothership is the navigation reference point for the Cubesats. It obtains its position with respect to Earth from observation, and ground tracking. There will be times when the Earth is not visible from the Mothership's position, so it will use extrapolation and local observation. During these periods of occultation, and for long one-way light times, the Mothership assumes local responsibility for the Health and Safety of the Swarm members. For this, we will implement Control Center functionality within the Mothership. This will take the form of a modified Ball Brother's COSMOS software. This product addresses traditional system test, integration, and flight needs. An additional software module is needed, essentially a virtual Control System Operator. Using

defined rules, the Mothership will make decisions concerning the Swarm Members, to the best of its current knowledge. All of this will be documented and downloaded to the Earth-based control center when communications is re-established. An AI capability will be added to Cosmos, in the form of a virtual flight controller agent. Besides the housekeeping functions, we will implement onboard science planning, responsive to on-site conditions, and targets of opportunity.

The Mothership's primary responsibility is continuance of the Mission. To a degree, the individual Cubesats are considered expendable. During communications black-outs, observations will continue, and the Mothership will dispense explorers according to pre-defined rules, and based on it's best on-scene judgment. It will also continue to collect observation science data, and engineering data related to health and performance across the swarm members.

This is a spin-off of an early NASA Project, called ANTS.

ANTS

Biological swarms, such as ants, achieve success by division of labor throughout the swarm, collaboration, and sheer numbers. They have redundancy, as any individual can do any task assigned to the swarm. The individual units are highly autonomous, but are dependent on other members for their needs. They achieve success with a simple neural architecture and primitive communications.

NASA did a lot of work in the 1970's and 1980's on ANTS – the Autonomous Nano-Technology Swarm. The Basic concepts were defined, but the implementation was not yet ready. Now, Cubesat-type architectures can address that. Actually, ANTS was preceeded by a Project called BEES, Biologically Inspired Engineering for Exploration Systems, as an architecture. The overall concept is many small units working together, self-configuring, self-healing, making in-situ decisions. No direct human control, not per-

programmed, but agile, making decision on the spot. A reference mission to the Asteroid Belt in 2020 was defined. Well, here we are, where's the mission? (This is partially addressed in the authors book, “A Cubesat Swarm Approach for Exploration of the Asteroid Belt, originally a student project that got to a high TRL). Another feature the swarm exhibits is teaming, social interaction between individual units. ANTS was hard to implement with the hardware and software of the 1980's. It fits well into a Cubesat implementation.

In Swarm robotics, the key issues are communication between units, and cooperative behavior. The capability of individual units nodes not much matter; it is the strength in numbers. Ants and other social insects such as termites, wasps, and bees, birds, fish, and reindeer are models for swarm behavior. Self-organizing behavior emerges from decentralized systems that interact with members of the group, and the environment. Swarm intelligence is an emerging field, and swarm robotics is in its infancy. Co-operative behavior, enabled by software and intra-unit communications has been demonstrated.

We can postulate target selection according to mission goals, but also mention that mission goals change as data is collected at the site.

Application Approaches

In this section we will discuss some of the concepts to enable the Icy Moons mission, and how these can be implemented in the Cubesat architecture. First, we are talking about deploying hundreds if not thousands of Cubesat explorers. In the paper study of Cubesats as an alternative to the Juno Spacecraft to Jupiter, within the size and weigh parameters, we found we could accommodate one thousand Cubesats. On the bus, or engineering side, they are all identical. They do carry different instrumentation. They can use different mobility platforms for different

environments, on land, in the water, under the water, or in the atmosphere. In some case, a deployer module will be used to transport the explorer-bots, It will also be a sensor platform, and a communications relay.

During the cruise phase, the Cubesats are unpowered. Every day or week (tbd), the units are powered on, one at a time, and checked for functionality. The onboard database is updated as required. The results are sent back to the control center on Earth. One advantage of the Mothership is, like the Shuttle, payloads can be tested before deployment. Known bad units will be left in place, discarded , or marked in the database as bad.

Failure in engineered systems is a function of complexity, the number of parts, and the number of interactions between parts. This works for both hardware and software. Here we are proposing to replace 1 or 2 large explorers with 1,000 smaller units. Although this seems to increase complexity, the commonality of the units stabilizes this. In addition, the system will be self-checking, and the Mothership will maintain a database of faults and failures, for trending information.

All of the the explorers are carried to the vicinity of the desired area of study by the Mothership, which itself uses a Cubesat architectural concept. It will enter a stable orbit, verify everything is operational, and begin to deploy relevant explorers, based on local conditions.

Communications

Several approaches to communication with spacecraft at large distances from Earth, and examining other planets, have been defined. The Interplanetary Internet implements a Bundle Protocol to address large and variable delays. Normal IP traffic assumes a seamless, end-to-end, available data path, without worrying about the physical mechanism. The Bundle protocol addresses the cases

of high probability of errors, and disconnections. This protocol was tested in communication with an Earth orbiting satellite in 2008. It is derived from the mobile data protocols used with cell towers in terrestrial applications. The Disruption Tolerant Network approach is also a good candidate.

There will be limited bandwidth, due to the distance from Earth. This was the case with APL's Horizon's spacecraft at Pluto – It took more than 16 months to transmit all the data back from the encounter.

The deployed Cubesats will use the Mothership and transporter as their communication relay, and not necessarily implement Cubesat-Cubesat direct communications. They do not necessarily need to implement a delay tolerant protocol, since the Cubesats will be “in the vicinity” of the Mothership. Cubesats will exchange three types of data with the Mothership: primary observational data, and secondary metadata which includes position, localization information, and timing information, and housekeeping on the explorer unit as well. All data will be kept and transmitted in CCSDS format.

Communication methodology within the network.

ISL (Inter-satelite link) communication, if required, will be achieved on Ultra High Frequency (UHF) links. An inter-satellite communication range of 90km to 100km is viable on UHF within the power output range of 4-5 W. The next challenge is regarding selection of the ideal antenna and communication protocol, keeping in mind the existing power and mobility constraints along with the tradeoff between radio power and communication distance. NASA’s Nodes (Network & Operation Demonstration Satellite) mission, similar in structure to the Edison Demonstration of Smallsat Networks (EDSN) mission, deployed a satellite swarm of Cubesats from the ISS to test inter-satellite communication capabilities in 2015. A primary UHF radio was used for crosslink

communication, and a further UHF beacon radio was used for transmitting real time health information of the satellite. In addition to this, position, navigation, and tracking information complement the primary data load. The Cubesats, and the Mothership will use software defined radio, implemented on the flight computer.

The Mothership will do onboard processing, as deep as possible, but always maintain the raw data in a database for a tbd period of time. This depends on the amount of solid state, radiation-protected memory the Mothership can host.

While we would like to transmit all of the raw observation data, we will be restricted by bandwidth and black-out times. Thus, some science data will be processed by the cluster on the mothership, and sent back to Earth via the Deep Space Net. Data processing onboard spacecraft, once a highly controversial issue (and a technical challenge), is now fairly routine.

NASA defined data processing levels, starting in 1986. These definitions are now mostly universally accepted. There are six levels:

NASA Data Processing Levels Definition

0 - Reconstructed, unprocessed instrument and payload data at full resolution, with any and all communications artifacts (e. g., synchronization frames, communications headers, duplicate data) removed.

1a - Reconstructed, unprocessed instrument data at full resolution, time-referenced, and annotated with ancillary information, including radiometric and geometric calibration coefficients and georeferencing parameters (e. g., platform ephemeris) computed and appended but not applied to the Level 0 data (or if applied, in a manner that level 0 is fully recoverable from level 1a data).

1b - Level 1a data that have been processed to sensor units (e. g., radar backscatter cross section, brightness temperature, etc.); not all instruments have Level 1b data; level 0 data is not recoverable from level 1b data.

2 - Derived geophysical variables (e. g., ocean wave height, soil moisture, ice concentration) at the same resolution and location as Level 1 source data.

3 - Variables mapped on uniform spacetime grid scales, usually with some completeness and consistency (e. g., missing points interpolated, complete regions mosaic-ed together from multiple orbits, etc.).

4 - Model output or results from analyses of lower level data (i. e., variables that were not measured by the instruments but instead are derived from these measurements).

An issue in data processing of observational data is the reversibility. Some operations on the data will mean we are no longer able to get back to the raw sensor input. If we have non-reversible operations, and the processing flow is incorrect, we have a problem since we can't access the raw data.

The Cubesat Explorers

The Cubesats will be a mix of 3U and 6U in size (a U-Unit is 30x10x10 cm), and follow the GSFC-defined PiSat architecture with a Raspberry-Pi flight computer, running NASA/GSFC's CFS/CFE flight software. Different instrumentation will be included on different Cubesats, using common platforms and buses. They will be deployed by the Mothership as required, to observe and collect data on targets of interest.

The Transporter

If we wish to explore different domains , a Transporter will be used to deliver the Cubesat Explorers. The design of the Transporter will depend on the target, but it will have the same Cubesat architecture. It is responsible for deploying the Explorers on the surface. For “water worlds” it will have the characteristics of a boat. For under-ice exploration it will have and added drilling capability, and will deploy submersible units. It will receive communications from the explorer, and serve as a store and forward node. The Transporter will be able to deploy flying explorers, if the atmosphere is dense enough. Multiple Transporters will be provided, each handling a mission-dependent number of explorers. In addition, the Transporter will serve as a navigation beacon for the explorers, using data from the in-orbit Mothership. The Cubesat explorers will not necessarily be retrieved, unless they have collected samples. The mechanism for a sample return has not yet been defined.

The Mothership

In this concept, the Cubesats are the primary payload. The Mothership can be thought of as a very large Cubesat. The architecture is kept as close as possible. It is intended that the mothership stays in orbit.

The Mothership will be built with standard aerospace products with a mission heritage. We expect to be able to use the same batteries and solar panels from the Juno mission, since the Mothership will be roughly the same size, defined by the size of the launch vehicle shroud. The X-band transceiver on Juno would be a candidate for the Earth link. The carrier is designed to be modular and adaptable. It uses standard PPOD Cubesat deployment mechanisms, oriented radially. Standard p-pod Cubesat dispensers are baselined, but the affects of long term storage of the Cubesats in the dispensers in space must be carefully considered. The effects of cold welding during the transit and

storage time needs to be evaluated.

Unlike Earth Cubesat missions, the Cubesats going to the Icy Moons will use the Transporter, which includes propulsion. The big limiting factor for them is electrical power. They can't carry large solar arrays. Dispersed from the carrier fully charged, they will operate as long as they can. The electronics and software will be optimized to minimize power usage. More advanced solar arrays, possible fabric-based arrays that will serve double duty as a solar sail, could solve this limitation.

The Mothership provides cloud services to the swarm. It is a store-and-forward node, and the communications relay to Earth. It provides Swarm control, monitoring, and task assignment.

The Mothership will have propulsion for orbit and cruise adjustments, and a propulsion system for attitude control and reaction wheel momentum unloading.

Avionics Suite

The Mothership, the Transporter, and the Explorers will baseline the GSFC PiSat software and hardware architecture for the flight computers. The Cubesats will use a single unit, and the Mothership will have a 16-unit cluster. Non-deployed Cubesats in the Mothership will be able to participate in the clustering, using the Mothership's internal networking infrastructure. The Mothership will be able to power up and attach selected additional units for particularly computer-intensive tasks.

The Raspberry Pi-3 is a very capable processor. An earlier model was tested to operate to 150 kRad Total Ionizing Dose, with only the loss of several unused I/O interfaces. The major source of radiation at the destination will not necessarily be trapped particles, but also ionizing cosmic rays of galactic origin. These are energetic, but sparse. The cluster computer will be enclosed in the

radiation vault located in the nose of the mothership.

A Raspberry Pi-3, baselined as the Cube-X onboard computer, requires 3.26 watts of power. It is quad core, operating at 1.4 GHz. It is a 64-bit machine, with 1 gigabyte of ram, and can achieve 2451 MIPS. It has a dedicated Graphics Processing Unit-based video pipeline that can handle 2D DSP. That is supported by the Open-GL software.

Compute cluster of convenience

Using a variation of the Beowulf clustering software and the communications infrastructure of the Mothership, the Cubesats awaiting deployment can be linked into a Compute cluster configuration. Each compute node will have the Beowulf software pre-loaded as part of its Linux operating system.

The Beowulf cluster is ideal for sorting and classifying data; an example application is the Probabilistic Neural Network. This algorithm has been used to search for patterns in remotely sensed data. It is computationally intensive, but scales well across compute clusters. It was developed by the Adaptive Scientific Data Processing (ASDP) group at NASA/GSFC. The program is available in Java source code.

Beowulf was developed to provide a low cost solution to linking commodity pc's into a supercomputer. The approach has been applied to clusters of small architectures such as those that serve as flight computers for Cubesats. Several 64-node Pi clusters have been demonstrated in the Earth environment.

The first Beowulf cluster to be flown in space was built from twenty 206-MHz StrongARM (SA1110) processors, and flew on the X-Sat, Singapore's first satellite. The performance was 4,000 MIPS. The cluster drew 25 watts. The satellite was a 100 kg unit, 80 CM cube. The cluster was used because the satellite collected large amounts of image data (80 GB per day), most of which was not relevant to the mission. An onboard classification algorithm

selected which images would be downloaded. For example, cloudy images and those over water were discarded, since land images of Singapore were of interest.

In a cluster, there is always a trade-off of computation and communication, with power draw. This can be monitored and adjusted by the cluster itself.

Swarm

This section describes a different approach to exploration with collections of smaller co-operating systems that will combine their efforts and work as ad-hoc teams on problems of interest.

Swarm behavior is based on the collective or parallel behavior of homogeneous systems. It is implemented as multiple co-operating software agents. This mimics collective behavior, modeled on biological systems. Examples in nature include migrating birds, schooling fish, and herding sheep. A collective behavior emerges from interactions between members of the swarm, and the environment. The resources of the swarm are organized dynamically. Swarm behavior is a group attribute across all members of the swarm. It relies on continuous co-ordination of behavior. Each member makes stochastic choices. There is a lack of authority in the Swarm, each member making local choices based on local conditions, and a global rule set. Example rules include separation, avoid crowding; alignment, do what other swarm members are doing; and cohesion, move to the center of mass of the swarm. Swarm activity can be chaotic or orderly. Emergent, un-planned behaviors are seen in implemented systems.

The capability of individual units does not much matter; it is strength in numbers. Self-organizing behavior emerges from decentralized systems that interact both with members of the group, and the environment. Swarm intelligence is an emerging field, and swarm robotics is an active research topic.

The cluster will be agile, able to respond to targets of opportunity, when they arise. Flexibility, and the ability to respond locally to unplanned events will be part of the architecture. A Swarm is an implementation of a multi-agent system, where the agents are implemented in program code.

A constellation of 50, 2U and 3U Cubesats was deployed in orbit in 2015. Some were released from the ISS, in coordination with a rocket launch. They collected and telemetered data on the lower thermosphere. This was not a swarm per se, but rather 50 units acting on their own, reporting back to their home institutions. Universities around the world participated in the project.

The data came from the region below 85 kilometers, which has enough of an atmosphere to impede spacecraft. The Cubesats collected data as long as they could, as they were reentering the atmosphere. At these altitudes, the rarefied atmosphere can reach temperatures of 2,500 degrees C. It is also a region where the dynamics are controlled by atmospheric tides, themselves controlled by diurnal heating and cooling. The member Cubesats used onboard processing to reduce downlink bandwidth.

Ops concept

During the cruise phase to the Jupiter and Saturn, the Cubesats are unpowered. Every week (tbd), the units are powered on, one at a time, and checked for functionality. The onboard database is updated as required. The results are sent back to the control center on Earth. One advantage of the Mothership is, like the Shuttle, payloads can be tested before deployment. Dead units will be left in place, unpowered, or discarded into Space, and noted in the database.

Near the desired target locations the Mothership uses its main

engine to enter planetary orbit with the solar panels oriented to the Sun, and the high gain antenna pointed to the Earth.

After another system check of itself and the Cubesats, the Mothership deploys a series of Cubesat scouts on a reconnaissance mission, to seek out areas of interest.

The Mothership deploys Cubesats with broad spectral sensing capabilities. Based on their findings, the Mothership may deploy additional Cubesats with specific instrumentation to the area of interest. (For example, an advanced thermal imager to an area of observed thermal activity). The Cubesats are released in the order of necessity. There will only be one Cubesat per dispenser, so blocking is not an issue.

In the surface exploration scenario, the lander necessarily needs a propulsion system for alignment and touchdown. This would be a cold gas system. It would be possible, but more complex, to implement a sample return. This would involve ascent and return to the Mothership, and rendezvous and docking. It is anticipated that the entire Cubesat with its payload would be returned to Earth, which may be simpler than a sample hand-off to a separate return vehicle. The sample would remain in Earth orbit (at the ISS), or at the Lunar Orbital Platform – Gateway in quarantine until it is deemed safe to send to Earth.

Overview of Explorer usage

Rovers have been used on Earth, the Moon, and Mars for exploration. In most cases, these have been autonomous, due to communication delays due to distance. Some underwater drones on Earth are tele-operated. The next challenges will be exploration of the diverse moons of Jupiter and of Saturn. Then there's the asteroid belt between Mars and Jupiter (1.9 million, larger than 1 km). These numbers are subject to change.

Teams of non-homogeneous co-operating robots can optimize the search for data or items of interest. A ground-based rover can be vectored to a position of interest by an autonomous air vehicle. Swarms of small robots cover larger areas, and come together as required for items of interest. Smaller vehicles can be deployed from larger platforms as well. This might include dropping a ground vehicle from an air vehicle, or deploying a smaller land vehicle from a larger platform.

The technologies developed for planetary exploration can be applied at Earth. Extreme environments on Earth are not well understood or documented. On Earth, we have the advantage of infrastructure that can support exploration systems, such as GPS for positioning, and satellite-based communication with low delays. Not to mention we have an atmosphere we can fly in.

Robots can operate autonomously, on their own, or via teleoperation, with a person in the loop. With communication delays of more than several seconds or so, teleoperation becomes difficult for the human operator. The best autonomous robots are operating on Mars. They receive top-level goals from Earth, and carry out those goals on their own. This requires a much greater sophistication.

The next mission to Mars in 2021 will include the 2020 Rover, which has a robotic helicopter. It will be an eye-in-the-sky, looking out for hazards, planning a path, and seeing things that the rover's camera can't. It will be autonomous in operation. It is a technology demonstration, planned to fly five times, during the early mission. The copter blades are a meter in diameter, and it has two counter-rotating sets. Compasses can't work on Mars due to the low magnetic field, so it will use solar tracking and inertial guidance. It will have its own solar panels. It is carried under the rover. It is dropped to the ground, and the rover moves some distance away so it can ascend. It runs the linux operating system.

Afterword

Not only is there strength in numbers, but an exploring swarm has a huge advantage on single or dual spacecraft. It comes more area, with more sensors, and support simultaneous observations from different points of view. The systems can make real time decisions on conditions observed at the target, and can handle targets of opportunity. It is quite feasible to send 1,000 thousand Cubesats to Jupiter and Saturn, based on the weight and size of the Juno mission. This is not a new concept, but has been defined and refined since the 1970's This represents a paradigm shift to the exploration of complex target systems. There are so many targets at Jupiter and Saturn, and we have to think that some may harbor life. Let's go see.

Glossary of terms and acronyms

Actuator – device which converts a control signal to a mechanical action.
A/D, ADC – analog to digital converter.
Analog – concerned with continuous values.
APL – (JHU) Applied Physics Lab)
Apojove – the point furthest away from Jupiter by a body that orbits it.
AR&D – Autonomous Rendezvous and Docking.
ASIN – Amazon Standard Inventory Number
Async – asynchronous; 2 processes not sharing the same clock.
BIST – built-in self test.
Bus – data channel, communication pathway for data transfer.
CAN – controller area network.
Centaur – a minor planet in an unstable orbit, behaving like an asteroid or comet.
Chip – integrated circuit component.
Clock – periodic timing signal to control and synchronize operations.
Comet – a solar system object consisting of ice, dust, and gas, in highly eccentric orbit
COSMOS (Ball Brothers) Open Source Control Center Software.
CPU – central processing unit.
Cube-X – a small mobility platform, mounting a Cubesat-based hardware and software architecture.
Cryovolcanism – volcano that spews out volatiles, such as water or methane.
DBMS – data base management system.
Device driver – specific software to interface a peripheral to the operating system.
DOF – degree of freedom.
DOI – digital object identifier.
Droid – robot.
DSP – digital signal processing
DTM – digital terrain model.

Ecliptic – the apparent path that the Sun seems to follow, the same as the Earth's orbit.
EDS – Electronic Data Sheet..
EMF – electro-magnetic field or force;.
ESA – European Space Agency
FPGA – field programmable gate array.
FRAM – ferromagnetic random access memory.
Gas giant – a large planet consisting mostly of hydrogen and helium. Jupiter and Saturn.
Giga - 10^9 or 2^{30}.
GHz – giga (10^9) hertz.
GNFIR - GSFC Natural Feature Image Recognition System.
GPR – ground penetrating radar.
GPS – global positioning system.
GPU – graphics processing unit.
Gray - unit of radiation, =100 rad
GSFC – Goddard Space Flight Center, Greenbelt, Maryland. NASA Center for unmanned spacecraft near Earth.
Hz – Hertz, or cycles per second.
IAU – International Astronomical Union.
Icy Moon – Moon consisting mostly of ice.
IEEE – Institute of Electrical and Electronic Engineers. Professional organization and standards body.
Interrupt – an asynchronous event to signal a need for attention example: the phone rings).
IP – Intellectual Property.
IR – infrared, 1-400 terahertz. Perceived as heat.
ISBN – International Standard Book Number.
ISS – International Space Station.
JHU – Johns Hopkins University.
Jovian – pertaining to Jupiter.
JPL – (NASA) Jet Propulsion Lab
L4 – Lagrangian point, null point in the gravitation field
L5 – Lagrangian point, null point in the gravitation field
LORAN – World War -II era radio navigation system
LRO – (NASA) Lunar Reconnaissance Orbiter.
LV – launch vehicle.

MRO – Mars Reconnaissance Orbiter.
NASREM - NASA/NBS Standard Reference Model for Telerobot Control System
NASA – National Aeronautics and Space Administration (USA)
NBS - National Bureau of Standards, now NIST.
NIST – National Institutes of Standards and Technology, formerly, National Bureau of Standards (NBS).
Open-GL – opensource graphics library,
Open source – methodology for hardware or software development with free distribution and access.
PAM – Prospecting Asteroid Mission.
Perijove – the point in an orbit closest to Jupiter by an object that orbits it.
PPOD - Poly-Picosatellite Orbital Deployer.
PSP – platform support package
RCS – robot control system.
RFI – Request for Information.
Ring system – a disk of solid material around a planet.
ROS – Robot Operating System (software)
RTOP - Research and Technology Objectives and Plans
RTOS – real-time operating system.
SARA – Saturn Autonomous Ring Array.
Schema – structure of a data base, expressed in a formal language.
Sensor – a device that converts a physical observable quantity or event to a signal.
Serial – bit by bit.
SSN – (DARPA) sub-surface navigation.
System – a collection of interacting elements and relationships with a specific behavior.
System of Systems – a complex collection of systems with pooled resources.
TDRSS – Tracking and Data Relay Satellite.
Telerobot – a robotic system with a human in the loop.
Transceiver – receiver and transmitter in one box.
TRL – Technology Readiness Levels.
Trojan - minor planet that shares an orbit with one of the larger planets.

Watchdog – hardware/software function to sanity check the hardware, software, and process; applies corrective action if a fault is detected; fail-safe mechanism.

XML - Extensible Markup Language.

References

Alvarez, Jennifer L.; Rice, John R. Samson, Jr., Michael A. Koets. "Increasing the Capability of Cubesat-based Software-Defined Radio Applications," avail: ieeexplore.ieee.org/document/7500847/

Ardila, David R. “Cubesats for Astrophysics,” Aerospace Corp. avail: https://cor.gsfc.nasa.gov/copag/rfi/CubesatsforAstrophysics.pdf

Baisamo, James M. et al “CubeSat technology adaption for in-situ characterization of NEOs,” presentation, avail: NASA Technical Reports Server (NTRS), 2014, document id 20140004799.

Challa, Obulapathi N., McNair, Janise “Distributed Data Storage on Cubesat Clusters, Advances in Computing 2013, (3)3 pp.36-49 Electronics and Computer Engineering, U. Florida, Gainsville.

Challa, Obulapathi N., McNair, Janise “Cubesat Torrent: Torrent-like distributed communications for Cubesat satellite clusters,” Military Communications Conference, 2012 , (MILCOM 2012) July 19, 2016. avail: https://www.researchgate.net/.../261237135

Clark, P. E. et al “BEES for ANTS: Space Mission Applications for the Autonomous NanoTechnology Swarm,” AIAA, Sept. 2004. avail: https://www.researchgate.net/publication/265158396_BEES_for_ANTS_Space_Mission_Applications_for_the_Autonomous_NanoTechnology_Swarm.

Cudmore, Alan Pi-Sat: A Low Cost Small Satellite and Distributed Mission Test Program, NASA/GSFC Code 582, avail: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150023353.pdf

Cudmore, Alan NASA/GSFC's Flight Software Architecture: core Flight Executive and Core Flight System, NASA/GSFC Code 582.

Curtis, S. A. et al "Use of Swarm Intelligence in Spacecraft Constellations for the Resource Exploration of the Asteroid Belt," 2003, Third International Workshop on Satellite Constellations and Formation flying, Pisa, Italy.

Dubowsky, S. et al "A Concept Mission: Microbots for Large-Scale Planetary Surface and Subsurface Exploration," AIP Conference Proceedings 746, 1449 (2005); https://doi.org/10.1063/1.1867276.

Fortescue, Peter and Stark, John *Spacecraft System Engineering*, 2nd ed, Wiley, 1995, ISBN 0-471-95220-6.

Gilster, Paul "CubeSats: Deep Space Possibilities," Sept. 2015, avail: http://www.centauri-dreams.org/?p=34056.

Hall, John "maddog"; Gropp, William *Beowulf Cluster Computing with Linux*, 2003, ISBN -0262692929.

Hinchey, Michael G. ; Rash, James L.; Truszkowski, Walter E.; Rouff, Christopher A., Sterritt, Roy *Autonomous and Autonomic Swarms,* avail: https://ntrs.nasa.gov/search.jsp?R=20050210015 2017-12-20T20:19:24+00:00Z.

McLoughlin, Ian; Bretschneider, Timo; Ramesh, Bharath "First Beowulf Cluster in Space," Linux Journal, September 2005, Issue #137, article 8097.

Mueller, Robert P. "Space Environment & Planetary Civil Engineering Basics," NASA KSC. Avail: https://kiss.caltech.edu/workshops/3d/presentations/mueller.pdf

Muri, P., McNair, J. "A Survey of Communication Sub-systems for

Intersatellite Linked Systems and Cubesat missions", J. Comm., 7 (2012), pp. 290–308.

Popescu, Otilia, "Power Budgets for Cubesat Radios to Support Ground Communications and Inter-Satellite Links" by (Senior Member, IEEE), Department of Engineering Technology, Old Dominion University, Norfolk,
avail: https://ieeexplore.ieee.org/document/7964683/

NASA, "Hitchhiking Into the Solar System: Launching NASA's First Deep-Space Cubesats," avail: www.nasa.gov/exploration.

NASA, Systems Engineering Handbook, NASA/SP-2007-6105, Rev. 1. (available on Google Books, Amazon, and others).

Popescu, Otilia, "Power Budgets for Cubesat Radios to Support Ground Communications and Inter-Satellite Links" by (Senior Member, IEEE), Department of Engineering Technology, Old Dominion University, Norfolk avai;l: , https://ieeexplore.ieee.org/document/7964683/

Spangelo, et al "JPL's Advanced CubeSat Concepts for Interplanetary Science and Exploration Missions, Cubesat Workshop," 2015, California Institute of Technology, JPL.
https://digitalcommons.usu.edu/cgi/viewcontent.cgi?
article=3313&context=smallsat

Stakem, Patrick H. "Free Software in Space–the NASA Case," invited paper, Software Livre 2002, May 3, 2002, Porto Allegre, Brazil.

Staehle, Robert et al, "Interplanetary Cubesats: Opening the Solar System to a Broad Community at Lower Cost" Cubesat Workshop, 2011, Logan Utah. Avail:
https://www.nasa.gov/pdf/716078main_Staehle_2011_PhI_Cubesat.pdf

Stakem, Patrick H.; Rezende, Aryadne; Ravazzi, Andre “Cubesat Swarm Communications,” 2016.

Stakem, Patrick H.; Da Costa, Rodrigo Santos Valente; Rezende, Aryasdne; Ravazzi, Andre “A Cubesat-based alternative for the Juno Mission to Jupiter, 2017, available from the author, pstakem@jhu.edu.

Stakem, Patrick H., Kerber, Jonathas “Rad-hard software, Cubesat Flight Computer Self-monitoring, Testing, Diagnosis, and Remediation,” 2017, available from the author, pstakem1@jhu.edu.

Stakem, Patrick H., Martinez, Jose Carlos; Chandrasenan, Dr. Vishnu; Mitra, Yash, “A Cubesat Swarm Approach for Exploration of the Asteroid Belt,” July 2018, presented at the NASA Goddard Space Flight Center Planetary CubeSats Symposium.

Stakem, Patrick H., Valente DaCosta, Rodrigo Santos, Rezende, Aryadne, Ravazzi, Andre, Chandrasenan, Vishnu, “A Cubesat-based alternative for the Juno Mission to Jupiter,” 2017, Proceedings of the Flight Software Conference.

Stakem, Patrick H., Martínez, José Carlos, Chandrasenan, Vishnu, Mitra, Yash, “A Cubesat Swarm Approach for Exploration of the Asteroid Belt”, 2018, Proceedings of the GSFC Planetary Cubesat Symposium.

Stakem, Patrick H., “ARM-based Compute Cluster for in-situ sensed data processing,” 2019, Presented at the NASA Goddard Space Flight Center Planetary CubeSats Symposium.

Tan, Ying *GPU-based Parallel Implementation of Swarm Intelligence Algorithms*, 2016, 1st ed, Morgan Kaufmann, ISBN-978-0128093627.

Truszkowski, Walt; Hallock, Harold L.; Rouff, Christopher; Karlin,

Jay; Rash, Hinchey, Mike; Sterritt, Roy *Autonomous and Automatic Systems,: With Applications to NASA Intelligent Spacecraft Operations and Exploration* Systems, Monographs in System and Software Engineering, Springer, 009, ISBN 9781846282331.

Truszkowski, Walt; Clark, P. E.; Curtis, S.; Rilee, M. Marr, G. A*NTS: Exploring the Solar System with an Autonomous Nanotechnology Swarm*. J. Lunar and Planetary Science XXXIII (2002).

Truszkowski, Walt; Clark, P. E.;, Curtis, S.; Rilee, M. Marr, G. *ANTS: Exploring the Solar System with an Autonomous Nanotechnology Swarm.* J. Lunar and Planetary Science XXXIII (2002).

Violette, Daniel P. "Arduino/Raspberry Pi: Hobbyist Hardware and Radiation Total Dose Degradation, EEE Parts for Small Missions," GSFC, 2014, avail: https://ntrs.nasa.gov/search.jsp?R=20140017620.

Resources

Small Spacecraft Technology State of the Art, NASA-Ames, NASA/TP2014-216648/REV1, July 2014.

Core Flight System (CFS) Deployment Guide, Ver. 2.8, 9/30/2010, NASA/GSFC 582-2008-012.

"NOAA Space Weather Scale for Radio Blackouts". NOAA / Space Weather Prediction Center. 2005-03-01.

Cubesat Design Specification, Cubesat Program, California Polytechnic State University, avail.https://www.google.com/searchq=Cubesat+Design+Specification&ie=utf-8&oe=utf-8

www.icubesat.org, Interplanetary Cubesat Workshop.

Cubesat Concept and the Provision of Deployer Services, avail:https://eoportal.org/web/eoportal/satellite-missions/content/-/article/Cubesat-concept-1

NASA Systems Engineering Handbook, NASA SP-2007-6105. Avail:https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20080008301.pdf

http://sen.com/news/cubesats-set-for-new-role-as-planetary-explorers

https://solarsystem.nasa.gov

http://ssed.gsfc.nasa.gov/pcsi

www.planetary.org

JPL Small-body Database. Avail:https://ssd.jpl.nasa.gov/sbdb.cgi

https://pds-rings.seti.org/jupiter/

http://astronomy.swin.edu.au/cosmos/C/Centaurs

https://svs.gsfc.nasa.gov/

https://voyager.jpl.nasa.gov/mission/science/

http://www.iflscience.com/space/how-saturns-shepherd-moons-herd-its-rings/

http://www.airspacemag.com/space/cubesats-moon-mars-and-saturn-too-180952389/

http://phys.org/news/2015-11-cubesats-deep-space.html

Ball Aerospace Cosmos Product, www.cosmosrb.com

http://sen.com/news/cubesats-set-for-new-role-as-planetary-explorers

Wikipedia, various.

ANTS/SWARM references

Baldassari, James Christopher L. Kopec, Eric S. Leshay, Walt Truszkowski, David Finkel "Autonomic Cluster Management System (ACMS): A Demonstration of Autonomic Principles at Work," ECBS 2005:512-518.

Clark, P. E., Rilee, M. L. Curtis, S. A., Truszkowski, Walt, Marr, G., Cheung, C. , Rudisill, M. "Bees for Ants: Space Mission Applications for the Autonomous NanoTechnology Swarm," 2004, avail: Researchgate.net.

Clark, P. E., Rilee, M. L., Curtis, S.A. "Exploring with PAM: Prospecting ANTS Missions for Solar System Surveys," 2003, 34th Annual Lunar and Planetary Science Conference, March 17-21, 2003, League City, Texas, abstract no.1493

Curtis, S. A.; Rilee, M. L., Clark, P. E., Marr, G. C. "USE OF SWARM INTELLIGENCE IN SPACECRAFT CONSTELLATIONS FOR THE RESOURCE EXPLORATION OF THE ASTEROID BELT," Third International Workshop on Satelllite Constellations and Formation Flying, Pisa, Italy, 24-26, 2003.

Hinchey, Michael G., Rash, James L., Truszkowski, Walter E. "Autonomous and Autonomic Swarms," avail: https://ntrs.nasa.gov/

Hinchey, Michael G., Sterritt, Roy, Rouff, Chris, Swarms and Swarm Intelligence, IEEE Computer, V 40, Issue 4, April 2007, Consortium on Cognitive Science Instruction.

Johnson, Michael A., Beaman, Robert G., JMica, Joseph A. Truszkowski, Walter F., Rilee, Michael L., Simm, David E. "Nanosat Intelligent Power System Development," avail: https://ntrs.nasa.gov/

Hinchey, Mike; Rash, James; Truszkowski, Walt; Rouff, Christopher; Sterritt, Roy “You Can't Get There from Here! Large Problems and Potential Solutions in Developing New Classes of Complex Computer Systems,”. Conquering Complexity, 2012: 159-176.

Hinchey, Mike; Dai, Yuan-Shun; Rash, James; Truszkowski, Walt; Madhusoodan, Manish;: “Bionic Autonomic nervous system and self-healing for NASA ANTS-like missions.”SA 2007: 90-96

Rilee, Michael L., Stufflebeam, Robert, “ANTS, Autonmous Nanotechnological Swarm,”

Rouff, Christopher; Hinchey, Mike; Truszkowski,Walt; Rash James; “Experiences applying formal approaches in the development of swarm-based space exploration systems,”. STTT 8(6): 587-603 (2006)

Rouff, Christopher; Hinchey, Mike; Rash, James; Truszkowski, Walt; “Towards a Hybrid Formal Method for Swarm-Based Exploration Missions,” SEW 2005: 253-264.

Rouff, Christopher, Truszkowski, Walt; Rash, James; Hinchey, Mike “Formal Approaches to Intelligent Swarms.,” SEW 2003:51

Truszkowski, Walt “Prototype Fault Isolation Expert System for Spacecraft Control,” NASA/GSFC, 1984.

Truszkowski, Walt et al, “Nanosat Intelligent Power System Development,“ avail: https://ntrs.nasa.gov/

Truszkowski, Walt, et al “Autonomous and Autonomic Systems, “ 2017. avail: https://ntrs.nasa.gov

Truszkowski, Walt et al, A Survey of Formal Methods for Intelligent Swarms, 2004.

Truszkowski, Walt et al, “Next generation system and software architectures Challenges from future NASA exploration missions,” 2006, Elsevier,

Truszkowski, Walt, et al ”Progressive autonomy: a method for gradually introducing autonomy into space missions,” ISSE 1(2): 89-99 (2005).

Truszkowski, Walt, et al “Towards an Autonomic Cluster Management System (ACMS) with Reflex Autonomicity”. ICPADS (2) 2005: 478-482.

Truszkowski,Walt et al “NASA's Swarm Missions: The Challenge of Building Autonomous Software,” IT Professional 6(5): 47-52 (2004)

Truszkowski, Walt et al “Asteroid Exploration with Autonomic Systems.” ECBS 2004: 484-489

Rouff, Christopher Amy Vanderbilt, Amy; Hinchey, Mike; Truszkowski, Walt; Rash, James “Verification of Emergent Behaviors in Swarm-based Systems,” ECBS 2004: 443-448

Vanderbilt, Amy et al, “Properties of a Formal Method for Prediction of Emergent Behaviors in Swarm-Based Systems,” SEFM 2004: 24-33.

from:dblp.unitrier.de

If you enjoyed this book, you might also be interested in some of these.

Stakem, Patrick H. *16-bit Microprocessors, History and Architecture*, 2013 PRRB Publishing, ISBN-1520210922.

Stakem, Patrick H. *4- and 8-bit Microprocessors, Architecture and History*, 2013, PRRB Publishing, ISBN-152021572X,

Stakem, Patrick H. *Apollo's Computers,* 2014, PRRB Publishing, ISBN-1520215800.

Stakem, Patrick H. *The Architecture and Applications of the ARM Microprocessors,* 2013, PRRB Publishing, ISBN-1520215843.

Stakem, Patrick H. *Earth Rovers: for Exploration and Environmental Monitoring,* 2014, PRRB Publishing, ISBN-152021586X.

Stakem, Patrick H. *Embedded Computer Systems, Volume 1, Introduction and Architecture*, 2013, PRRB Publishing, ISBN-1520215959.

Stakem, Patrick H. *The History of Spacecraft Computers from the V-2 to the Space Station*, 2013, PRRB Publishing, ISBN-1520216181.

Stakem, Patrick H. *Floating Point Computation*, 2013, PRRB Publishing, ISBN-152021619X.

Stakem, Patrick H. *Architecture of Massively Parallel Microprocessor Systems*, 2011, PRRB Publishing, ISBN-1520250061.

Stakem, Patrick H. *Multicore Computer Architecture,* 2014, PRRB

Publishing, ISBN-1520241372.

Stakem, Patrick H. *Personal Robots*, 2014, PRRB Publishing, ISBN-1520216254.

Stakem, Patrick H. *RISC Microprocessors, History and Overview,* 2013, PRRB Publishing, ISBN-1520216289.

Stakem, Patrick H. *Robots and Telerobots in Space Application*s, 2011, PRRB Publishing, ISBN-1520210361.

Stakem, Patrick H. *The Saturn Rocket and the Pegasus Missions, 1965,* 2013, PRRB Publishing, ISBN-1520209916.

Stakem, Patrick H. *Visiting the NASA Centers, and Locations of Historic Rockets & Spacecraft,* 2017, PRRB Publishing, ISBN-1549651205.

Stakem, Patrick H. *Microprocessors in Space*, 2011, PRRB Publishing, ISBN-1520216343.

Stakem, Patrick H. Computer *Virtualization and the Cloud*, 2013, PRRB Publishing, ISBN-152021636X.

Stakem, Patrick H. *What's the Worst That Could Happen? Bad Assumptions, Ignorance, Failures and Screw-ups in Engineering Projects, 2014,* PRRB Publishing, ISBN-1520207166.

Stakem, Patrick H. *Computer Architecture & Programming of the Intel x86 Family, 2013,* PRRB Publishing, ISBN-1520263724.

Stakem, Patrick H. *The Hardware and Software Architecture of the Transputer*, 2011,PRRB Publishing, ISBN-152020681X.

Stakem, Patrick H. *Mainframes, Computing on Big Iron*, 2015, PRRB Publishing, ISBN- 1520216459.

Stakem, Patrick H. *Spacecraft Control Centers*, 2015, PRRB Publishing, ISBN-1520200617.

Stakem, Patrick H. *Embedded in Space,* 2015, PRRB Publishing, ISBN-1520215916.

Stakem, Patrick H. *A Practitioner's Guide to RISC Microprocessor Architecture*, Wiley-Interscience, 1996, ISBN-0471130184.

Stakem, Patrick H. *Cubesat Engineeering*, PRRB Publishing, 2017, ISBN-1520754019.

Stakem, Patrick H. *Cubesat Operations*, PRRB Publishing, 2017, ISBN-152076717X.

Stakem, Patrick H. *Interplanetary Cubesats*, PRRB Publishing, 2017, ISBN-1520766173 .

Stakem, Patrick H. Cubesat Constellations, Clusters, and Swarms, Stakem, PRRB Publishing, 2017, ISBN-1520767544.

Stakem, Patrick H. *Graphics Processing Units, an overview*, 2017, PRRB Publishing, ISBN-1520879695.

Stakem, Patrick H. *Intel Embedded and the Arduino-101, 2017,* PRRB Publishing, ISBN-1520879296.

Stakem, Patrick H. *Orbital Debris, the problem and the mitigation*, 2018, PRRB Publishing, ISBN-*1980466483.*

Stakem, Patrick H. *Manufacturing in Space*, 2018, PRRB Publishing, ISBN-1977076041.

Stakem, Patrick H. *NASA's Ships and Planes*, 2018, PRRB Publishing, ISBN-1977076823.

Stakem, Patrick H. *Space Tourism*, 2018, PRRB Publishing, ISBN-

1977073506.

Stakem, Patrick H. *STEM – Data Storage and Communications*, 2018, PRRB Publishing, ISBN-1977073115.

Stakem, Patrick H. *In-Space Robotic Repair and Servicing*, 2018, PRRB Publishing, ISBN-1980478236.

Stakem, Patrick H. *Introducing Weather in the pre-K to 12 Curricula, A Resource Guide for Educators*, 2017, PRRB Publishing, ISBN-1980638241.

Stakem, Patrick H. *Introducing Astronomy in the pre-K to 12 Curricula, A Resource Guide for Educators*, 2017, PRRB Publishing, ISBN-198104065X.
Also available in a Brazilian Portuguese edition, ISBN-1983106127.

Stakem, Patrick H. *Deep Space Gateways, the Moon and Beyond*, 2017, PRRB Publishing, ISBN-1973465701.

Stakem, Patrick H. *Exploration of the Gas Giants, Space Missions to Jupiter, Saturn, Uranus, and Neptune*, PRRB Publishing, 2018, ISBN-9781717814500.

Stakem, Patrick H. *Crewed Spacecraft*, 2017, PRRB Publishing, ISBN-1549992406.

Stakem, Patrick H. *Rocketplanes to Space*, 2017, PRRB Publishing, ISBN-1549992589.

Stakem, Patrick H. *Crewed Space Stations,* 2017, PRRB Publishing, ISBN-1549992228.

Stakem, Patrick H. *Enviro-bots for STEM: Using Robotics in the pre-K to 12 Curricula, A Resource Guide for Educators,* 2017, PRRB Publishing, ISBN-1549656619.

Stakem, Patrick H. *STEM-Sat, Using Cubesats in the pre-K to 12 Curricula, A Resource Guide for Educators*, 2017, ISBN-1549656376.

Stakem, Patrick H. *Lunar Orbital Platform-Gateway*, 2018, PRRB Publishing, ISBN-1980498628.

Stakem, Patrick H. *Embedded GPU's*, 2018, PRRB Publishing, ISBN- 1980476497.

Stakem, Patrick H. *Mobile Cloud Robotics*, 2018, PRRB Publishing, ISBN- 1980488088.

Stakem, Patrick H. *Extreme Environment Embedded Systems,* 2017, PRRB Publishing, ISBN-1520215967.

Stakem, Patrick H. *What's the Worst, Volume-2*, 2018, ISBN-1981005579.

Stakem, Patrick H., *Spaceports*, 2018, ISBN-1981022287.

Stakem, Patrick H., *Space Launch Vehicles*, 2018, ISBN-1983071773.

Stakem, Patrick H. *Mars*, 2018, ISBN-1983116902.

Stakem, Patrick H. *X-86, 40th Anniversary ed*, 2018, ISBN-1983189405.

Stakem, Patrick H. *Lunar Orbital Platform-Gateway*, 2018, PRRB Publishing, ISBN-1980498628.

Stakem, Patrick H. *Space Weather*, 2018, ISBN-1723904023.

Stakem, Patrick H. *STEM-Engineering Process*, 2017, ISBN-1983196517.

Stakem, Patrick H. *Space Telescopes,* 2018, PRRB Publishing, ISBN-1728728568.

Stakem, Patrick H. *Exoplanets*, 2018, PRRB Publishing, ISBN-9781731385056.

Stakem, Patrick H. *Planetary Defense*, 2018, PRRB Publishing, ISBN-9781731001207.

Patrick H. Stakem *Exploration of the Asteroid Belt*, 2018, PRRB Publishing, ISBN-1731049846.

Patrick H. Stakem *Terraforming*, 2018, PRRB Publishing, ISBN-1790308100.

Patrick H. Stakem, *Martian Railroad,* 2019, PRRB Publishing, ISBN-1794488243.

Patrick H. Stakem, *Exploiting the Moon,* 2019, PRRB Publishing, ISBN-1091057850.

Patrick H. Stakem, *RISC-V, an Open Source Solution for Space Flight Computers,* 2019, PRRB Publishing, ISBN-1796434388.

Patrick H. Stakem, *Arm in Space*, 2019, PRRB Publishing, ISBN-9781099789137.

Patrick H. Stakem, *Extraterrestrial Life*, 2019, PRRB Publishing, ISBN-978-1072072188.

Stakem, Patrick H. *Riverine Ironclads, Submarines, and Aircraft Carriers of the American Civil War*, 2019, PRRB Publishing.

Stakem, Patrick H. *Submarine Launched Ballistic missiles*

Stakem, Patrick H., *Space Command, Military in Space, 2091*

2020 Releases

CubeRovers, a synergy of Technologys

Exploration of Lunar & Martian Lava Tubes by Cube-X

Hacking Cubesats

History and Future of Cubesats